Проект
«Создание программной платформы для взаимодействия разработчиков с документами»

Автор: Андрей Е. Шевель (shevel.andrey@gmail.com)
Вариант-Дата: 2025-10-09 (на основе прежних вариантов)

Аннотация

Целью проекта является разработка системы искусственного рецензента для автоматизация когнитивного процесса разработки текста документа. Предполагается, что документ из технической области, где все утверждения документа должны технически соотноситься друг с другом на непротиворечивой основе, а все или большинство утверждений документа могут быть проверены объективным образом. Документ может быть статьёй, отчётом, описанием нового технического проекта, руководства администратора сложной установки и т.д. Первоначальный вариант документа для рецензирования представляет разработчик.

В отличие от рецензента (человека) предлагаемая система искусственного рецензента позволяет разработчику текста улучшать разрабатываемый текст в цикле взаимодействия с системой не ограничивая число циклов взаимодействия, что может приводить к новым идеям в кругу разработчиков (в естественных нейронных сетях, т.е. голове разработчика). Новые идеи могут оказаться много важнее очередной версии документа.

В настоящее время имеется немало вариантов применения методов машинного обучения в техническом рецензировании научных статей, поступающих в научно-технические журналы. Отметим важное отличие искусственного рецензента от рецензента человека. Рецензент человек даже если он использует технику машинного обучения как правило выполняет рецензирование, чтобы представленный текст удовлетворял определённым стандартам по содержанию, объёму, другим параметрам, включая ограниченное время на принятие решения – принять текст к публикации или отвергнуть. Предлагаемая система искусственного рецензирования нацелена на помощь разработчику, не накладывая на разработчика формальных ограничений, которые не имеют отношения к содержательной части текста. Взаимодействуя с системой искусственного рецензирования, разработчик волен значительно изменить содержание или внести совершенно новое содержание. Также разработчик текста сам решает показать кому-либо свой вариант текста или нет.

Будучи реализованной, система позволит работать с текстами документов ограниченного распространения. По завершении цикла улучшения документа разработчиком, представляющего собой инструкцию по эксплуатации сложной технической установки, вариант системы вместе с документом можно использовать как помощник администратору этой установки.

Основное назначение предлагаемой системы выполнять роль технического помощника разработчика в улучшении текста документа разработчиком. Основные решения в отношении документа принимает разработчик, а не предлагаемая система.

Введение

При разработке любого документа заметного размера (статья, отчёт, проект) часто возникает необходимость неоднократного уточнения формулировок и состава разделов. Особенно актуальна такая необходимость, когда имеются ограничения на распространение содержания документа за пределы коллектива авторов до завершения разработки документа.

Если нет ограничений на распространение содержания документа до его завершения, то можно воспользоваться одним из приглашённых рецензентов, который может помочь разработчикам обратить внимание на форму изложения и/или использованную терминологию. Примерно также можно поступить, использовав в качестве рецензента одну из Больших Языковых Моделей[1] (БЯМ), доступных в Интернет, которые могут быть загружены в локальную сеть на один из локальных серверов. Эту БЯМ можно использовать на другом сервере из той же локальной сети, но без связи с Интернет. Предполагается, что загруженная БЯМ уже где-то хорошо тренирована на большом объёме текстов. Локально БЯМ планируется использовать в рамках архитектуры Поиска и Дополнительной Генерации[2] (ПДГ), которая включает средства взаимодействия с системой на основе вэб приложения фронтенд[3] (frontend). В данной схеме для функционирования БЯМ не требуется контакт с Интернет.

Нередко разрабатываемый документ имеет ограниченное распространение, то практически исключены использование любых Интернетовских сайтов для помощи в рецензировании и улучшения документа. Более того, на сервере, где будет происходить разработка и рецензирование документа, должно быть в целях безопасности гарантировано отсутствие прямых каналов связи с Интернет.

Разработка документа может содержать десяток и более страниц, а возможно и несколько сотен страниц. В дальнейшем термин разработчик будет использовать как для отдельного человека, так для группы разработчиков. Документ с ограниченным

---

[1] Большая языковая модель (БЯМ; англ. large language model, LLM) — языковая модель, состоящая из нейронной сети со множеством параметров (обычно миллиарды весовых коэффициентов и более), обученной на большом количестве неразмеченного текста с использованием обучения без учителя. БЯМ может генерировать релевантное продолжение введённого текста.

[2] Генерация, дополненная поиском (Retrieval-Augmented Generation, RAG) — это подход, при котором генерация ответа большой языковой модели (LLM) осуществляется на основе данных, полученных в результате поиска во внешних источниках (файлы, базы данных, Интернет и другие источники). RAG-система работает в два основных этапа: сначала происходит извлечение релевантных документов или их частей из внешней базы знаний на основе запроса пользователя, а затем полученная информация подставляется вместе со специальными подсказками, указывающими как модель должна использовать эти данные, в контекст языковой модели для генерации итогового ответа. В зависимости от указаний в подсказках, сгенерированный ответ может включать цитаты или ссылки на исходные документы, что повышает прозрачность и доверие пользователей, позволяя проверить информацию. RAG помогает устранить ограничения LLM, такие как устаревание информации, наличие неточностей или появления галлюцинаций

[3] Фронтенд (Frontend) — это та часть веб-сайта или приложения, которую видит и с чем взаимодействует пользователь в браузере. Он отвечает за создание визуальной части (интерфейса), в которую входят кнопки, текст, изображения, формы и анимации, а также за интерактивность, делающую сайт удобным для пользователя.

распространением готовится и рецензируется на локальном сервере. Очень полезно иметь такой сервер и такое программное обеспечение, которое могло бы поддерживать взаимодействие между разработчиками и меняющимся в процессе разработки текстом документа.

Взаимодействие должно быть организовано таким образом, чтобы очередной версии документа можно было задать вопрос посредством фронтенда, использование архитектуры ПДБ, где генерация выполняется посредством подобранной БЯМ и получить ответ строго на основании содержания версии документа с указанием номеров страниц документа, т. е. для функционирования БЯМ не требуется контакт с Интернет. Вопросы задаются письменно на естественном языке. Ответы также выдаются письменно на таком же естественном языке или, если задано, на другом естественном языке. Для проверки такой системы полезно подготовить несколько десятков или больше вопросов и оценить ответы системы, например по шкале от 0 до 10. Оценка выполняется разработчиком, который сличает ответы системы с текстом разрабатываемого документа. Таким образом производится сертификация правильности ответов архитектуры ПДБ самим разработчиком документа.

Возможно, первый вариант рецензируемого документа, представленный в систему, приведёт к неверным ответам системы по содержанию документа и, естественно к низким оценкам ответов системы разработчиком. Поскольку система в архитектуре ПДБ использует для анализа текущую версию текста документа, то разработчику логично анализировать текст документа в тех местах, где имели место недостаточно хорошие ответы системы. В частности, среди прочего, следует проверить ответ системы на вопрос «Всё ли достаточно точно и полно изложено в документе как необходимо?». Иными словами, ответы системы определяется текстом версии документа. После очередного редактирования документа разработчиком с целью улучшения содержания документа получается очередная версия текста документа. Теперь потребуется заново ввести во фронтенд те же и/или дополнительные вопросы и снова получить ответы системы. Разработчик должен снова произвести оценку полученных ответов.

Повторяя цикл несколько раз, внося корректировки и дополнения в текст и получая очередную версию текста, разработчик будет обдумывать направление коррекций, что полезно во всех отношениях. Важным достоинством является факт, что во время обсуждений у разработчика могут возникнуть новые идеи по поводу содержания документа, которые окажутся значительно более ценными, чем небольшие уточнения текста.

Такого вида систему назовём Платформа Искусственный Ассистент (ИА) или кратко - ПИА. Таким образом, ПИА есть программная система, реализованная в архитектуре ПДБ, которая включает в себя поисковую подсистему, локальный образ БЯМ, вэб фронтенд, хранилище для версий документов.

Реализация

Для реализации ПИА на основе ПДБ архитектуры потребуется сервер с необходимыми параметрами и программным обеспечением. Удобней всего реализацию проекта выполнять под управлением ОС Линукс опираясь свободно распространяемое программное обеспечение с открытыми исходными кодами. Сам сервер должен быть обеспечен оперативной памятью 128+ GB. Процессор или два не старше 1-2 лет типа Intel 2-3 GHz. Дисковая память на SSD ёмкостью 2-6 TB будет достаточной на первых порах.

Важным элементом являются наличие GPU приличной ёмкости (не менее 80 GB на GPU). Число GPU также важно для скорости выполнения архитектуры ПДБ, рекомендуется использовать два GPU. Прежний опыт показал, что БЯМ с числом параметров меньше 32В, дают не очень точные результаты. Таким образом следует использовать БЯМ с числом параметров не менее 32В. С такими БЯМ для быстрого (~1 минута) ответа потребуется один или два GPU с памятью не менее 80GB каждый. Также многое зависит от объёма документов. В основном имеются в виду тексты документов проектов, статей, описаний и т.д. суммарного объёма около ста страниц и более. Если суммарный рамер обрабатываемых документов вырастет до многих тысяч страниц, то потребуются более мощные компьютерные установки.

Алгоритм использования ПИА

В данном разделе описан *алгоритм использования ПИА* или *цикл ПИА*. Алгоритм состоит из следующих шагов, часть из которых выполняет разработчик. Возможно, часть шагов алгоритма может выполняться автоматически. Зелёным цветом выделены операции, которые должен выполнять разработчик (человек).

1. Запустить программный фронтенд для взаимодействия с ПИА.
2. Преобразовать текст документов в векторную базу данных. Преобразование выполняется специальными программными средствами под общим названием embedding[4].
3. Подготовить или скорректировать список проверочных вопросов.
   3.1. Число вопросов должно соответствовать сложности документа, например 3 вопроса на страницу для небольших документов возможно достаточно. Для крупных документов не исключены другие соотношения, которые определяются разработчиками документов.
4. Ввести проверочные вопросы посредством фронтенда в ПИА.
5. Получить ответы на вопросы.
6. Оценить ответы ПИА на введённые вопросы по шкале от 0 до 10.
7. Если имеются оценки ниже 10, то следует рассмотреть соответствующие разделы документов и постараться скорректировать текст документов таким образом, чтобы при следующем проходе оценка бы выросла. Однако, окончательная оценка качества документа остаётся за разработчиком. Предлагаемая система является только техническим помощником разработчика.
8. Если хотя бы одни ответ ниже некоторого предела, который должны определить разработчики, то документы следует скорректировать.

[4] Термин embedding (эмбеддинг) — это метод технологии машинного обучения, который преобразует многомерные сложные данные, такие как слова, изображения или узлы графа, в вектор чисел с плавающей точкой меньшей размерности (векторный эмбеддинг). Этот вектор сохраняет семантическое значение и взаимосвязи исходных данных, позволяя моделям машинного обучения обрабатывать и понимать их более эффективно для таких задач, как классификация текстов, семантический поиск и извлечение информации.

9. Если все оценки оказались выше установленного предела, то можно полагать, что задача выполнена. В противном случае следует перейти к пункту 2.

Общая схема функционирования ПИА представлена на рис. 1.

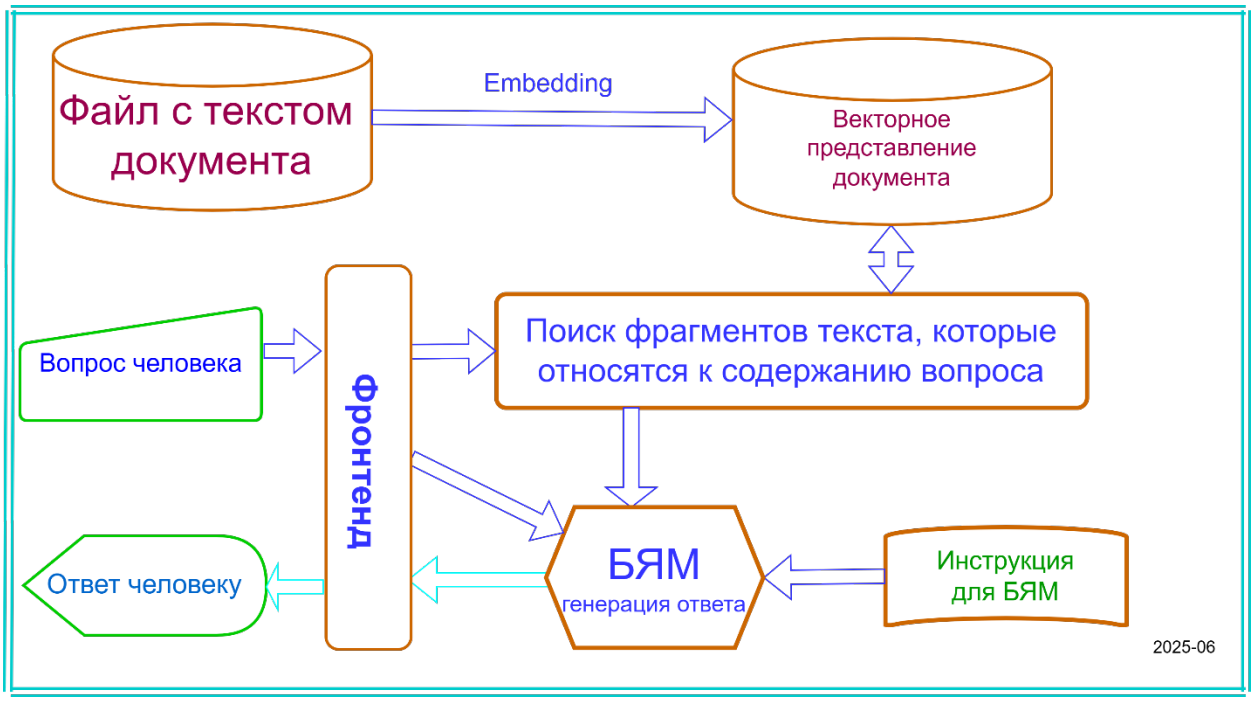

Рис. 1 Общая схема функционирования ПИА.

Пояснения к рисунку Рис. 1.

Текст документа вначале преобразуется в векторное представление (векторную базу данных). Человек разработчик вводит вопрос в окне фронтенда. Введённый вопрос поступает в подсистему «Поиск фрагментов текста, которые относятся к содержанию вопроса» Поиск осуществляется в векторной базе данных. Найденные фрагменты текста поступают на вход в БЯМ одновременно с текстом вопроса человека и с инструкцией что должна делать БЯМ. Результат генерации БЯМ передаётся человеку посредством фронтенда.

На рисунке 2 представлен цикл работы с текстом документа.

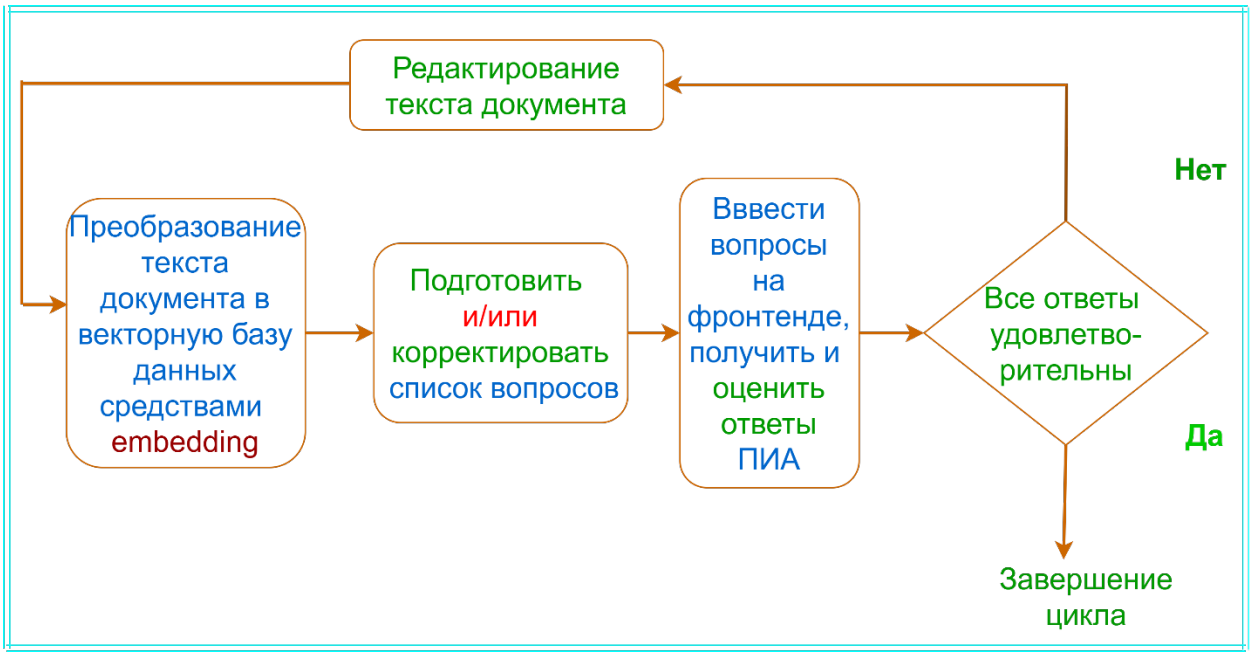

Рис. 2 Схема цикла ПИА.

Пояснения к рисунку 2.

Человек разработчик вводит вопросы из приготовленного списка вопросов с использованием фронтенда, как показано на рисунке 1. Ответы ПИА должны быть проанализированы и оценены человеком. На рисунке зелёным выделены действия или решения, которые должен принимать человек, т.е. разработчик. После рецензирования или коррекции текста документа снова потребуется преобразовать уже скорректированный текст документа в векторную базу данных.

Оценки ответов ПИА

Ответы ПИА должны выполняться разработчиком. Предлагается использовать шкалу от 0 до 10. Примерный расклад оценок таков как показано в таблице 1.

Таблица 1 Пример определения оценок ответов ПИА.

Оценка ответа на вопрос	Причина	Замечание
0	Не имеет отношения к тексту.	ПИА должна информировать, что вопрос не релевантен, т.е. надо сформулировать другой вопрос.
1-3	Много (больше 6) неверных ответов ПИА.	Требуется редактировать текст в тех местах, где много ошибок таким образом, чтобы в дальнейших циклах ПИА число ошибок уменьшалось. Возможно добавление и/или реорганизация разделов.
4-6	4 и более неверных ответов ПИА.	Требуется редактировать текст в тех местах, где много ошибок таким образом, чтобы в дальнейших циклах использования ПИА число ошибок уменьшалось. Возможно как добавление и реорганизация разделов, так и уточнения формулировок и терминов.
7-10	Каждый неверный ответ логично приводит к уменьшению оценки разработчика на 1.	Полагать ли, что один или больше неверных ответов ПИА есть приемлемый результат – это выбор разработчиков, поскольку ПИА есть только технический помощник разработчика. Тем не менее рекомендуется уточнение текста с целью уменьшения числа ошибок в следующих циклах использования ПИА Не исключаются появление новых идей в отношении содержания документа, что даст бОльший положительный эффект.

ПИА позволяет выявить неудачные части текста, что помогает разработчику улучшить текст документов. Если с введением коррекций в тексте появляются новые формальные несоответствия в документе, которые ПИА выявляет, то скорее всего разработчик документа не совсем ясно и полно изложил в документе свои идеи. Иными словами, ПИА не может заменить разработчика, а служит только техническим помощником для улучшения текста документа самим разработчиком.

Потенциальные сценарии использования такой платформы

- Сценарий подготовки описания проекта сложной технической системы
  - Использование цикла ПИА позволит уменьшить время ввода нового дополнительного разработчика, поскольку новый разработчик сможет оперативно получать ответы на свои вопросы о деталях разработки.
  - Использование цикла ПИА поможет на ранних этапах разработки обнаружить противоречия, излишнее дублирование, а также уточнить формулировки.
  - Значительным положительным фактором является обсуждение внутри команды (если их более одного) разработчиков ответов ПИА, которое весьма вероятно породит новые идеи в отношении полезных деталей разработки.
    - ***Замечание***: число циклов ПИА при работе над описанием проекта может отражать активность (или прогресс) в подготовке проекта.
- Сценарий подготовки статьи, отчёта
  - Использование цикла ПИА поможет найти противоречия в тексте, например путём ввода вопроса «Какие противоречия обнаружены в документах?». Каждое найденное потенциальное противоречие должно понижать оценку ответа на 1. Разработчику путём редактирования текста документа следует добиваться устранения подозрения на противоречие в документе.
  - Подготовка релевантного списка содержательных вопросов, получения и оценки ответов также позволит уточнить места в документе, где полезно улучшить текст следующим образом:
    - добавление новых пояснений к сложным разделам;
    - уточнение терминологии для лучшего понимания;
    - убрать дублирующие части текста.
  - ПИА может использоваться в качестве рецензента, который позволяет уточнить формулировки и используемые термины.
- Сценарий подготовка описания технической установки, например сложного компьютерного кластера.
  - Цикл ПИА может значительно помочь разработчикам в улучшении качества и полноты описания с использованием цикла улучшения.
- Сценарий использование ПИА в качестве справочного средства по содержанию документов, например для пользователей сложной установки/программы.
  - Например, администраторы сложной технической системы могут воспользоваться ПИА как источником дополнительной информации к обычной документации.

- ПИА может ответить на вопросы по описанию технической установки. Вопросы могут быть, например такими «Что можно предпринять для уменьшения загрузки блока 12?» или «Где найти информацию по работе блока 12?»

Выше показано, что в разных сценариях использования ПИА способна помогать в ряде аспектов. Кратко формулируя достоинства использования ПИА состоит в следующем:

- Сокращение времени разработки описаний больших документов (описаний проектов или сложных систем, книг, статей) с использованием цикла ПИА.
  - Вопросы к документам в рамках системы ПИА готовятся на естественном языке (в данном случае на Русском языке).
  - Ответы системы ПИА по содержанию документов также на естественном языке (в данном случае на Русском языке).
  - Использование системы ПИА позволяет ускорить добавление нового разработчика.
  - Фиксируемое число циклов ПИА может использоваться как метрики прогресса в составлении текста описания или отчета, статьи.
- Использование системы ПИА позволяет ускорить добавление новых администраторов сложных технических систем, а также упрощает обучение новых администраторов.

Требования к создаваемой платформе

ПИА должна обеспечивать относительно лёгкую настройку путём изменения простых конфигурационных параметров. Настройка должна включать следующее:

- Тип БЯМ; в проекте предполагается использовать открытые свободно распространяемые БЯМ, большинство из которых вместе с описаниями находятся по ссылке [1,2].
- Параметры БЯМ (около 10).
- Название каталога, где хранятся документы для работы (PDF).
- Возможность поиска в документах по строкам на основе grep и/или bm25.
- Средства формирования ответов в виде файлов PDF в заданном каталоге.

Сервер, на котором установлено программное обеспечение, не должен иметь прямого контакта в Интернет. Необходимое программное обеспечение в том числе для целей модернизации должно поставляться через сетевой шлюз с другого локального сервера, который имеет доступ в Интернет. Отсутствие прямого контакта в Интернет никак не влияет на производительность ПИА.

Оценки затрат на создание

В данном разделе описаны оценки финансовых и трудозатрат для реализации компьютерной установки ПИА. Отметим, что обсуждается установка, на которой хранятся и обрабатываются документы ограниченного распространения, которые не могут быть размещены или обработаны в публичной облачной системе. Оценки затрат следующие

- Сервер 128 GB, CPU Intel, 4 TB SSD, 2 x GPU A100 ~3700К рублей (~45.8% от суммарных затрат);
    - Инженерные компоненты: кабели, источник бесперебойного питания (ИБП) ~1000К (~12.4% от суммарных затрат);
    - Чем больше внутренняя память GPU и выше его производительность, тем выше производительность установки в целом.
- Подготовка места для размещения оборудования, размещение оборудования и подключение кабелей, проверка функционирования оборудования, установка операционной системы, установка прикладного программного обеспечения, комплексное тестирование – примерно 2 человеко-месяца[5] ~260К рублей (3.2% от суммарных затрат).
- Детальная разработка архитектуры, подбор и тестирование программного обеспечения, подготовка и проведение Программы и Методики Испытаний (ПМИ), подготовка описания, обучение пользователей – 24 человеко-месяца ~3120К рублей (38.6% суммарных затрат).
- План-график разработки и запуска прототипа – 12 месяцев.
- Итого финансовые затраты оцениваются как ~ 3700К + 1000К + 260К + 3120К = 8080К рублей.

Установка ПИА может использоваться предположительно несколькими разработчиками (до 3 в данной конфигурации), что увеличивает отдачу от инвестиций в проект.

Потенциальный риск с задержкой реализации проекта может быть обусловлен задержкой с поставкой оборудования.

Ответственность за использования ПИА

ПИА используется разработчиком. Ответы генерируемые ПИА зависят от ряда аспектов подготовки всей системы.

Во-первых, очень многое зависит от квалификации разработчика, который используют ПИА. Сама ПИА помогает разработчикам решать технические задачи, она не генерирует новые содержательные соображения, которых нет в документах. Новые соображения в содержание текста может добавить только разработчик, т.е. человек.

Естественно, что администратор сам отвечает за решение вопроса «Использовать ПИА или не использовать ПИА». Естественно, что администратор отвечает за решение вопроса «Последовать рекомендации ПИА или не последовать рекомендации ПИА». Эту

[5] Примерная оценка трудозатрат один человеко-месяц оценивается в данном документе как размер зарплаты в месяц ~130К рублей.

ситуацию легко сравнить с навигатором в автомобиле – водитель отвечает за ДТП, а не навигатор.

Очевидно, что ПИА будет выдавать верные ответы на вопросы администратора или предлагать варианты решений, если в ПИА загружены тщательно подготовленные документы описаний. В этом случае получение ответа ПИА на вопрос администратора будет заметно быстрее, чем если администратор будет вручную искать ответы среди сотен страниц описаний.

Ожидаемый эффект от внедрения ПИА

Затраты времени на разработку содержательной части крупных документов предположительно значительно сократится по сравнению с обращением за помощью к реальному человеку-рецензенту. Оценки сокращения времени могут быть следующими. Если человек рецензент может подготовить рецензию на крупный документ вряд ли быстрее, чем за неделю, то разработчик может использовать ПИА и редактировать текст документа 5 раз за неделю или чаще без привлечения человека-рецензента. Таким образом можно ожидать уменьшение времени на разработку большого документа в 5 или более раз при наличии квалифицированного разработчика. Такое сокращение времени на разработку отчётов, проектов и т.п. документов также значительно повысит производительность организации, где используется ПИА, в целом, что приведёт в годовом исчислении увеличения прибыли (или экономии) больше суммы затрат на реализацию данного проекта. В конечном итоге основй целью ПИА является увеличение производительности разработчика.

Ссылки

1. Сайт https://huggingface.co/
2. Открытые модели https://github.com/eugeneyan/open-llms

Конец текста